HAW HAMBURG

Data Mining Cup 22

Ein grober Einblick in Aufgabe und Vorgehen

INHALT

1 – SZENARIO & TASK

Das Szenario:
Ein Onlineshop versendet als Marketing-Maßnahme regelmäßig Newsletter, um Produkte zu bewerben

Das Problem:
Oft werden Produkte beworben, die von den Kunden gerade erst gekauft wurden und somit uninteressant sind

Die Aufgabe:
Gesucht wird ein Modell, das möglichst zuverlässig die Woche voraussagt, in der ein Kunde eines seiner am häufigsten gekauften Produkte erneut kauft
(hier am Beispiel des Monats Februar 2021)

***prediction**:*

***0** = kein Kauf im Februar*
***1** = Kauf in der ersten Februarwoche (01.02. – 07.02.2021)*
***2** = Kauf in der zweiten Februarwoche (08.02. – 14.02.2021)*
***3** = Kauf in der dritten Februarwoche (15.02. – 21.02.2021)*
***4** = Kauf in der vierten Februarwoche (22.02. – 28.02.2021)*

	userID	itemID	prediction
1	0	20664	NaN
2	0	28231	NaN
3	13	2690	NaN
4	15	1299	NaN
5	15	20968	NaN
...	...	...	...
9996	46118	20106	NaN
9997	46124	19677	NaN
9998	46125	12878	NaN
9999	46127	7963	NaN
10000	46130	395	NaN

2 – DIE DATEN

Auszug der Bestellhistorie vom 01.06.2020–31.01.2021

	date	userID	itemID	order	brand	feature_1	feature_2	feature_3	feature_4	feature_5	categories
1	2020-06-01	276	15667	1	1201	4	0	30	0	163	[1680, 813, 218, 3915, 3914, 4069]
2	2020-06-01	276	28708	1	504	10	0	441	3	84	[2591, 2312, 2708, 3603]
3	2020-06-01	532	7644	1	1276	6	0	45	3	48	[813, 327, 1390, 3915, 3914, 3920]
4	2020-06-01	752	22963	1	1201	10	0	43	0	147	[1456, 1986, 327, 3389, 747, 698, 3915, 3413, ...
5	2020-06-01	1123	18498	1	1401	4	0	95	0	44	[2178, 646, 644, 1463, 1390, 3915, 4019, 2096,...
...	...	...	...	...	...	...	...	...	...	...	...
39732	2021-01-16	8786	14420	1	703	4	0	335	0	132	[583, 1330, 3915, 3976]
39733	2021-01-17	19221	8927	2	745	10	0	503	0	85	[3150, 3503, 2995, 1694, 2863]
39734	2021-01-17	34638	4935	1	1127	4	0	360	3	144	[1760, 1259, 493, 1082, 3915, 3912, 3914, 1244...
39735	2021-01-20	21517	594	1	203	4	1	491	0	66	[1920, 3923]
39736	2021-01-20	21517	19443	1	408	10	0	160	0	38	[1871, 3228]

3 – FEATURE ENGINEERING (AUSZUG DER EINBEZOGENEN FEATURES)

Repeat Customer Probability
Repeat-Customers / (Onetimer + Repeat-Customers)
„25% aller Käufer von Item X haben es erneut gekauft“

Mean Difference To Next Purchase (User)
Dauer (Tage) bis zum Wiederholungskauf durch User

Mean Difference To Next Purchase (Item)
Dauer (Tage) bis zum erneuten Kauf eines Produkts

Total Item Orders (User)
Gesamte Produktkauf-Stückzahl durch User

Total Item Orders (Item)
Gesamte Produktkauf-Stückzahl aller User

Brand-Order-Ratio
Punktzahl zu Einordnung der Markenbeliebtheit
(bezogen auf alle Bestellungen der Gesamthistorie)

Feature-Order-Ratio (1-5)
Punktzahl zu Einordnung der Feature-Beliebtheit
(bezogen auf alle Bestellungen der Gesamthistorie)

Total-Brand-Feature-Score
Gesamtpunktzahl aus *Brand-* & *Feature-Order-Ratio*

Date(Year)
Jahr des Kaufdatums

Date (Month)
Monat des Kaufdatums

Date (Day Of Month)
Tag des Kaufdatums

Date (Week Of Year)
Woche im Jahr des Kaufdatums

Date (Day Of Year)
Tag im Jahr des Kaufdatums

Next Buy in Weeks (floor)
Wochendifferenz zum Wiederholungskauf

+ Brand, Features, Categories, Orders

4 – FEATURE ENGINEERING: KATEGORIEN TRANSFORMIEREN

Kategorien

- String mit variabler Länge
- Mit unterschiedlichen Kategorien (Range von 0-4299)

categories
[1680, 813, 218, 3915, 3914, 4069]
[2591, 2312, 2708, 3603]
[813, 327, 1390, 3915, 3914, 3920]
[1456, 1986, 327, 3389, 747, 698, 3915, 3413, ...
[2178, 646, 644, 1463, 1390, 3915, 4019, 2096,...
...
[583, 1330, 3915, 3976]
[3150, 3503, 2995, 1694, 2863]
[1760, 1259, 493, 1082, 3915, 3912, 3914, 1244...
[1920, 3923]
[1871, 3228]

Transformation:

- Umwandlung der Strings in eine Liste (String-Split)
- *Multi-Hot-Codierung* der Kategorien
  - → Eine Spalte pro Kategorie
  - → Kategoriezugehörigkeit eines Items wird Binär angezeigt
  - → Feature-Vektor ist so bei allen Items gleich groß

	ITEM	category
0	1	[0, 5, 7]
1	2	[2, 6, 7, 8]
2	3	[1, 6]
3	4	[2, 9, 4, 3, 7]
4	5	[6]

	ITEM	0	1	2	3	4	5	6	7	8	9
0	1	1	0	0	0	0	1	0	1	0	0
1	2	0	0	1	0	0	0	1	1	1	0
2	3	0	1	0	0	0	0	1	0	0	0
3	4	0	0	1	1	1	0	0	1	0	1
4	5	0	0	0	0	0	0	1	0	0	0

- Allerdings: massive Erhöhung der Dimensionen (4300 Kategorien)

4 – FEATURE ENGINEERING: KATEGORIEN TRANSFORMIEREN

Sparse Matrix (*Dünnbesetzte Matrix):*
Platzverschwendung durch Speichern vieler Nullen

0	1	2	3	4	5	6	7	8	9
1	0	0	0	0	1	0	1	0	0
0	0	1	0	0	0	1	1	1	0
0	1	0	0	0	0	1	0	0	0
0	0	1	1	1	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0

0	1	2	3	4	5	6	7	8	9
1					1		1		
		1				1	1	1	
	1					1			
		1	1	1			1		1
						1			

```
(0, 0) 1
(0, 5) 1
(0, 7) 1
(1, 2) 1
(1, 6) 1
(1, 7) 1
(1, 8) 1
(2, 1) 1
(2, 6) 1
(3, 2) 1
(3, 3) 1
(3, 4) 1
(3, 7) 1
(3, 9) 1
(4, 6) 1
```

→ **bspw. nur *non-zero-values* mit „Storage-by-Index“-Methode speichern**

5 – ANSÄTZE

1. **Data Exploration / Feature Engineering**
   a) RCP
   b) Baumstruktur in den Kategorien
2. **Time-Series**
3. **Unsupervised**
   a) K-Means Clustering
   b) Assoziationsanalyse
4. **Recommender System**
   a) Ähnlichkeiten von Käufen eines Kunden zu Käufen eines anderen Kunden erkennen.
   b) Datenlage nicht ausreichend
5. **Klassifikation**
   a) Labeling-Problem (stattgefundener Kauf)
   b) XGB, Decision Tree (Multi-Hot-Codierung, Sparse Matrix)
   c) Labelanpassung (zukünftige Käufe)
   d) Every-Day-Klassifikation (zu jedem der 273 Tage vom 01.06.–28.02. eine Reihe pro möglicher User-Item-Kombination generieren & labeln)*
   → stattgefundene Käufe Label *1-4*, andere *0*
6. **Regression**
   a) Modellvergleich (Linear Regression, XGBRegressor, Random Forest, ...)
7. **Differenz zur Standardabweichung**
   a) Vorhersage von Kunden, die eigentlich im Februar landen müssten, es aber mit der ersten Vorhersage nicht tun

* Nur Kombinationen der *submission.csv:* 273 Tage × 10000 Kombinationen = 2730000

6 – VERFOLGTE BETRACHTUNGSWEISEN

Zwei Betrachtungsweisen des Problems:

1. **Klassifikation**
   Bestimmung der Klassen {0, 1, 2, 3, 4, 5} für die von der Aufgabe geforderte Vorhersage der Februar-Woche (Kauf in Woche 1-4 oder 0 für *kein Kauf*)

Label?

- **Kaufdatum umwandeln in Woche des Monats**
  (Training Jun-Jan, dann Klassifiz. von Items für Feb)
  - → **Klassifikation** kann aber nur anhand bekannter Klassen vorgenommen werden
  - → wo kommt die Null her?
    (z.B. Käufe mit in Monate aufnehmen, die dort nicht stattgefunden haben, in anderen aber schon)
  - → Weiteres Problem: Zeitkomponente fehlt

2. Numerische Vorhersage
Prognose des nächsten Kaufzeitpunkts durch einen beliebigen numerischen Wert

Label?

- **Label = Dauer bis zum nächsten Kauf**
  (**Numerische Vorhersage** des nächsten Kaufs)
  - → Ein aktueller Kauf erhält einen Wert, der angibt, wie lange es bis zum nächsten Kauf dauert
  - → Problem: je näher ein Kauf am Ende des Datenset-Zeitraums (31.01.2021), desto mehr *Nullen* (wird nicht mehr gekauft) entstehen und werden *angelernt*

Aus der numerischen Vorhersage wird dann ein Datum ermittelt und zur Februar-Woche umgewandelt.

7 – MODELLTEST MIT DATENSPLIT

Split der Daten in Trainings- und Testset
Keinen zeitlichen Split (z.B. Jun-Dez & Jan) vornehmen, sondern:

Trainingsset:

- Enthalten alle Käufe außer letzten, damit der nächste Kauf immer bekannt ist
  → da Label ≠ 0, wird *0* nicht übermäßig antrainiert

Predictionset:

- Enthalten alle Käufe, zu denen das nächste Kaufdatum nicht bekannt ist (Label = 0), weil danach keiner mehr stattgefunden hat
- Produkte, die zum ersten Mal gekauft werden sind uninteressant, da *submission.csv* nur Produkte enthält, die mind. 2x gekauft wurden

Um **Modelle überprüfen** zu können, wurde der Datensatz stärker beschnitten:
Nur **User-Item-Kombinationen**, die **mind. 4x** auftraten wurden beibehalten. Die **letzten Käufe** wurden **entfernt** (können nicht überprüft werden, da kein Folgekauf)

Trainingsset:

- Enthält immer **mind. 2 gleiche User-Item-Kombinationen** (Käufe eines Kunden desselben Produkts)

Testset:

- Enthält immer den vorletzten Kauf, sodass sich Label (nächster Kauf in X Zeiteinheiten) überprüfen lässt

8 – VERGLEICH REGRESSIONSMODELLE (AUSZUG)

- Zunächst Klassifikation verfolgt, schließlich die **numerische Vorhersage mittels Regressions-Verfahren**
- Auszug der Ergebnisse von **Regressionsmodellen** auf dem stark beschnittenen (mind. 4 Käufe) Datensatz
- Auswahl des Modells anhand der **Performance auf dem Testdatenset**
- Hohe Trefferquoten auf dem Trainingsset aber verhältnismäßig schlechte auf dem Testdatenset weisen auf ***Overfitting*** hin (z.B. Random Forest)

XGBRegressor

	train	test
Row count of set:	39736	13591
Correctly predicted rows:	11057 (27.826 % of rows)	1516 (11.154 % of rows)
MSE:	5.363	16.979
MAE:	1.561	3.086

LGBMRegressor

	train	test
Row count of set:	39736	13591
Correctly predicted rows:	11345 (28.551 % of rows)	1367 (10.058 % of rows)
MSE:	5.678	17.084
MAE:	1.593	3.122

KNeighborsRegressor

	train	test
Row count of set:	39736	13591
Correctly predicted rows:	7096 (17.858 % of rows)	1673 (12.310 % of rows)
MSE:	8.586	16.528
MAE:	2.139	2.978

LinearRegression

	train	test
Row count of set:	39736	13591
Correctly predicted rows:	7418 (18.668 % of rows)	1968 (14.480 % of rows)
MSE:	7.605	16.129
MAE:	1.958	2.824

RandomForestRegressor

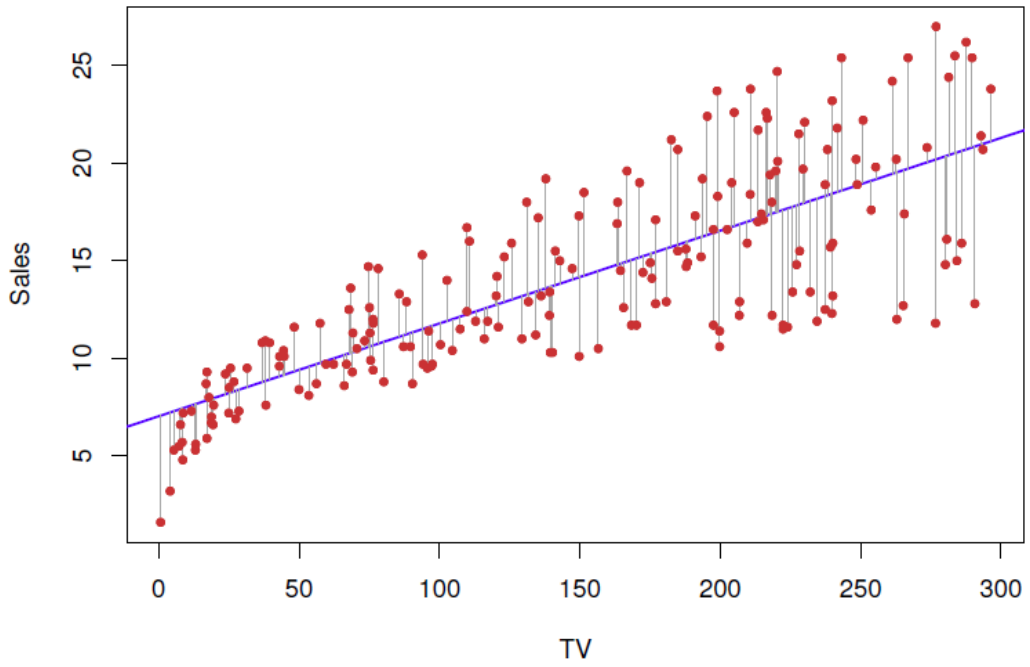
	train	test
Row count of set:	39736	13591
Correctly predicted rows:	21535 (54.195 % of rows)	1420 (10.448 % of rows)
MSE:	1.093	17.228
MAE:	0.618	3.141

Ridge

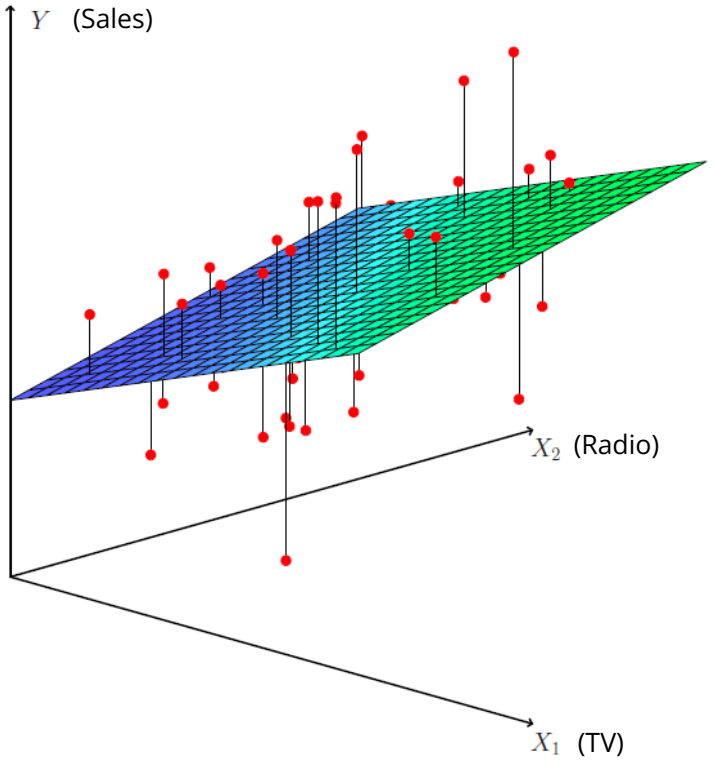
	train	test
Row count of set:	39736	13591
Correctly predicted rows:	7436 (18.714 % of rows)	1898 (13.965 % of rows)
MSE:	7.613	16.449
MAE:	1.956	2.86

9 – LINEARE REGRESSION

- Einfache Lineare Regression



- Multiple Lineare Regression mit 2 Variablen



Abbildungen: James, G., Witten, D., Hastie, T. & Tibshirani, R. (2021). *An Introduction to Statistical Learning: with Applications in R (Springer Texts in Statistics)* (2nd ed. 2021 Aufl.). Springer.

9 – LINEARE REGRESSION

- Einfache Lineare Regression

$$Y = \beta_0 + \beta_1 X + \epsilon$$

- Y: Zielgröße (erklärte Variable)
- X: Einflussgröße (erklärende Variable)
- ϵ : Stochastischer Fehler (z.B. Messfehler)

- Multiple Lineare Regression mit mehreren Variablen

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

Abbildungen: James, G., Witten, D., Hastie, T. & Tibshirani, R. (2021). *An Introduction to Statistical Learning: with Applications in R (Springer Texts in Statistics)* (2nd ed. 2021 Aufl.). Springer.

9 – LINEARE REGRESSION

- Lineares Regressionsmodell (Regressionskoeffizienten sind nicht bekannt)

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

- Formel, mit der wir versuchen eine Vorhersage zu machen, die die tatsächliche Funktion möglichst exakt beschreibt

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p$$

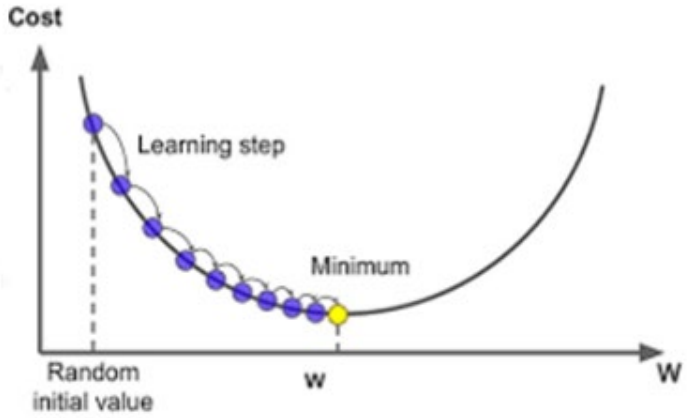
- ZIEL: Alle Regressionkoeffizienten β_p bestimmen, sodass die Summe aller Abweichungen zwischen unserem vorhergesagten Wert $\hat{y}_i$ und dem tatsächlichen Wert y_i möglichst gering ist

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

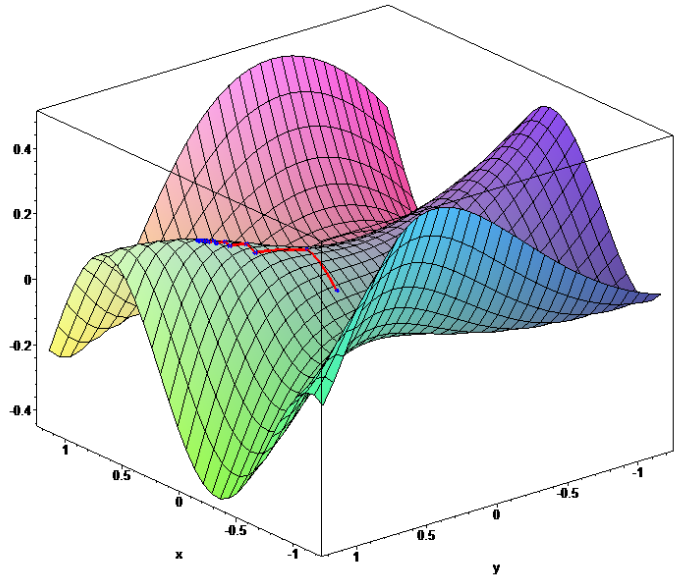
Abbildungen: James, G., Witten, D., Hastie, T. & Tibshirani, R. (2021). *An Introduction to Statistical Learning: with Applications in R (Springer Texts in Statistics)* (2nd ed. 2021 Aufl.). Springer.

9 – LINEARE REGRESSION

- Minimierung der Fehlerfunktion mittels Gradientenabstieg



$$
\begin{aligned}
&\text{repeat until convergence } \{ \\
&\quad \theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \\
&\quad \theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) \cdot x^{(i)} \\
&\}
\end{aligned}
$$



10 – UMGANG MIT PREDICTIONS VOR FEBRUAR

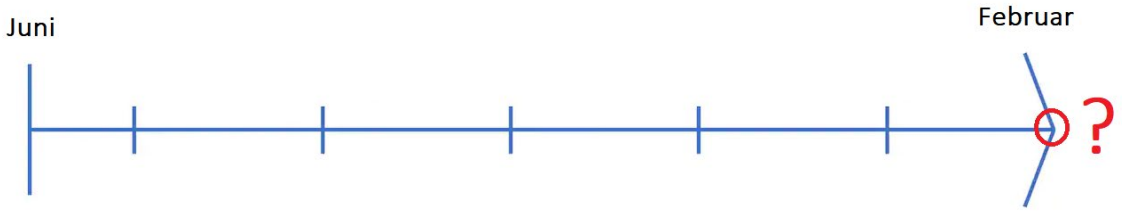
- 630 Predictions nicht im Februar gelandet, obwohl die Kunden das Produkt regelmäßig bestellt haben (min. 6x im ges. Zeitraum)
- Passiert, wenn Kd. regelmäßig kauft und das Intervall klein ist (z.B. alle 2 Wochen)

- **Problem:**
  - → Hat Kd. aufgehört zu kaufen (Fall 1) oder wird so regelmäßig kauft, dass Prediction vor Februar landet und eigentlich im Februar landen müsste, wenn man es weiterführt (Fall 2) ?
- **Erster Ansatz:**
  - → Neue Prediction mit der Prediction
- **Zweiter Ansatz:**
  - → Delta vom 31.01. bis zum letzten Kauf ausrechnen
  - → Wenn Delta kleiner als *½ * Standardabweichung + Mittelwert*, dann zur WeekOfYear den Mittelwert addieren

Fall 1

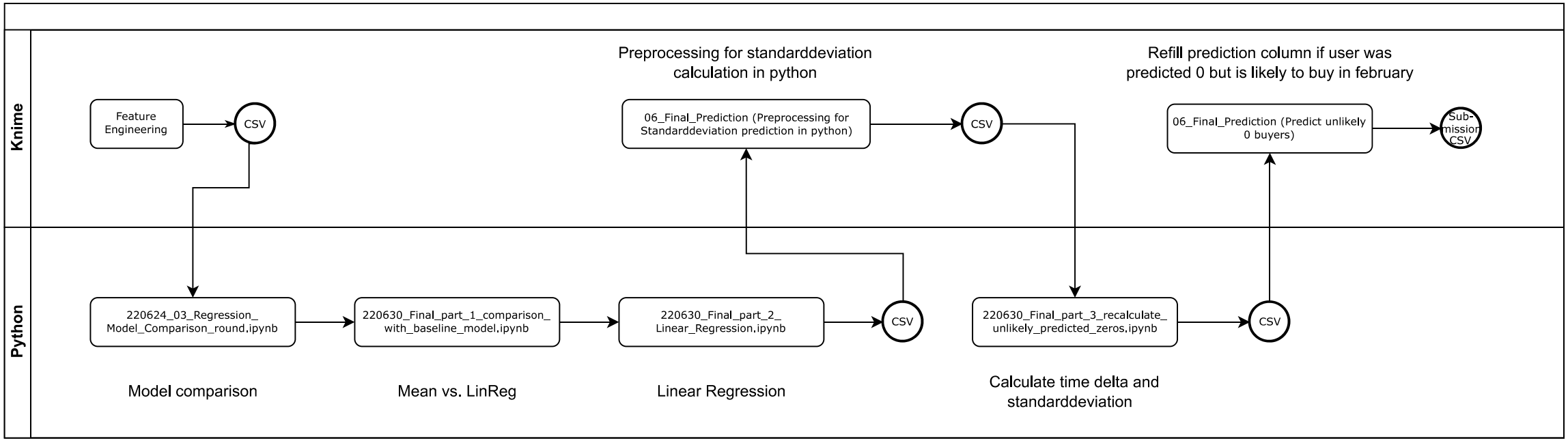
Kunde kauft wahrscheinlich nicht nochmal, Delta zum letzten Kauf zu groß

Fall 2



Kunde kauft wahrscheinlich, da Delta kleiner als ½ * Standardabweichung + Mittelwert

11 – WORKFLOW MIT KNIME & PYTHON



12 – HERAUSFORDERUNGEN

- **Rechenzeit**
  (nicht alle Verfahren rechnen mit der GPU, CSVs benötigen lange zum Einlesen/konvertieren)
- **Fehlender Speicherplatz**
  (zu wenig RAM und/oder VRAM für zu viele Rows/Dimensionen)
- **Model lernt zu viele „*Nullen*“ kennen**
  (Classifier & Regression)
- **Data Leakage**
  Falsche Datenselektion/-vorverarbeitung verrät dem Modell Informationen, die es nicht kennen dürfte (z.B. Mittelwert Zeitdifferenz von Produkten „aus der Zukunft“; Train hat Informationen über Testset)
- **Regression liefert nur das nächste Ergebnis**, benötigt sind aber Vorhersagen bis in den Februar hinein oder darüber hinaus

13 – LEARNINGS

- **Sinnvolles Selektieren und Labeln** von Daten hat enormen **Einfluss auf angewandte Verfahren** und umgekehrt
- Oft auch **technische Herausforderungen**, die **durch intelligentes Datenhandling** gelöst werden müssen
  (oder durch mehr Rechenpower)
- **Ansätze** möglichst schnell **im Kleinen vorab testen**, bevor die richtige Implementierung vorgenommen wird
- Verschiedene Ansätze möglichst **früh gegeneinander Testen**
- Noch viel eher **überprüfen, ob es sich überhaupt um ein Maschine-Learning-Problem handelt** oder einfache Methoden ähnlich gut funktionieren
- Datenhandling erfordert **extreme Konzentration**, um keine Fehler einzubauen
- **Statistische Grundlagen** sehr von Vorteil
- **Automatisch berechnete Metriken** zur Beurteilung von Modellen müssen mit Vorsicht genossen und **immer überprüft werden**

Vielen Dank

Kevin Kröll
Moritz Uhlig
Leander Wernst